Week-5

Aim:- to study different activation functions used in deep learning implement them and analyze their role in introducing Non-linearity improving learning capability

Algorithm:-

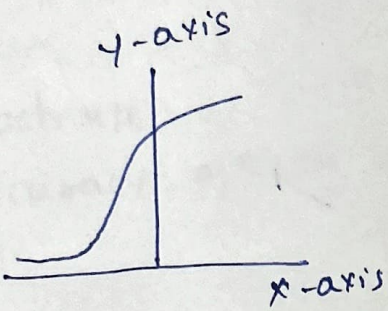
→ start
→ initialize the input values
→ define commonly used activation functions:-
  → sigmoid : $f(x) = \frac{1}{1+e^{-x}}$
  → Tanh : $f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$
  → Relu (rectified linear unit)
    $f(x) = \max(0, x)$
  → soft max : $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$
4, Apply each activation function on input data
5, plot / observe the output behaviour for different range of input
6, Train a small neural network with each activation function and compare performance
7, stop

pseudo code:

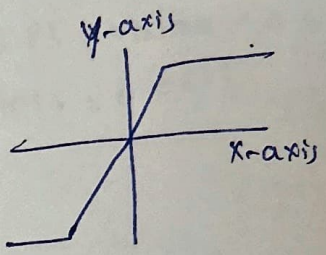
start
→ load MNIST dataset
→ Normalize image and flatten to vectors
→ for each activation in [sigmoid, tanh, relu]:
  - Build model with 2 hidden layers using that activation.
  - train model for 5 epochs
  - evaluate on test data.

~~DL~~

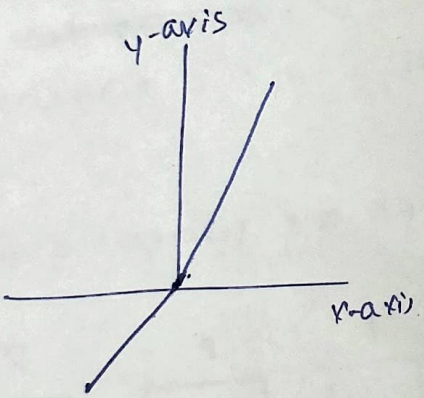
Sigmoid :- $g(x) = \frac{1}{1+e^{-x}}$



Tanh :-
$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

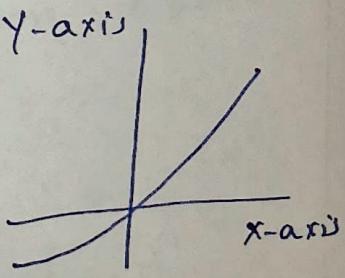


leaky relu :-

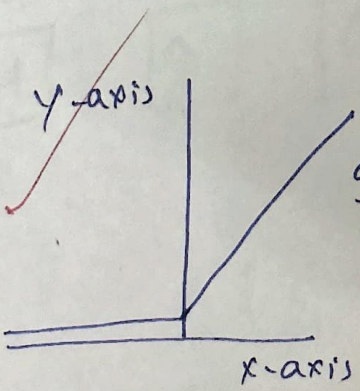


$g(x) = \max(0.1x, x)$

Exponential linear units :-



ReLU :-



$g(x) = \max(0, x)$

- Print accuracy and loss

END

observation:

1) the sigmoid activation function showed slower convergence and achieved lower accuracy compared to other

2) tanh performed better than sigmoid since it is zero centered, but still suffered from vanishing gradients

4) the choice of activation function had a significant effect on both training speed and final model accuracy

result: it succesfully implemented the activation functions used in deeplearning.

1.23/4

```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(-1, 784).astype('float32') / 255.0
x_test  = x_test.reshape(-1, 784).astype('float32') / 255.0
y_train = to_categorical(y_train, 10)
y_test  = to_categorical(y_test, 10)

for act in ['sigmoid','tanh','relu','elu']:
    model = models.Sequential([
        layers.Dense(128, activation=act, input_shape=(784,)),
        layers.Dense(64, activation=act),
        layers.Dense(10, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    model.fit(x_train, y_train, epochs=5, batch_size=128, verbose=0)
    _, acc = model.evaluate(x_test, y_test, verbose=0)
    print(f"Activation: {act:<7} | Test Accuracy: {acc:.4f}")
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
/usr/local/lib/python3.12/dist-packages/keras/src/layers/core/dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim` ar
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Activation: sigmoid | Test Accuracy: 0.9609
Activation: tanh    | Test Accuracy: 0.9733
Activation: relu    | Test Accuracy: 0.9746
Activation: elu     | Test Accuracy: 0.9740
```